Parameter Estimation of the Kinetic α-Pinene Isomerization Model Using the MCSFilter Algorithm

Andreia Amador[1], Florbela P. Fernandes[2], Lino O. Santos[1], Andrey Romanenko[1,3], and Ana Maria A. C. Rocha[4]

[1] CIEPQPF, Department of Chemical Engineering, Faculty of Sciences and Technology of the University of Coimbra, 3030-790 Coimbra, Portugal, `uc2010149251@student.uc.pt`, `lino@eq.uc.pt`
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, 5300 - 253 Bragança, Portugal, `fflor@ipb.pt`
[3] Ciengis, SA, Rua Pedro Nunes, Ed.E, Coimbra, Portugal, `andrey.romanenko@ciengis.com`
[4] Algoritmi Research Centre, University of Minho, 4710-057 Braga, Portugal `arocha@dps.uminho.pt`

Abstract. This paper aims to illustrate the application of a derivative-free multistart algorithm with coordinate search filter, designated as the MCSFilter algorithm. The problem used in this study is the parameter estimation problem of the kinetic α-pinene isomerization model. This is a well known nonlinear optimization problem (NLP) that has been investigated as a case study for performance testing of most derivative based methods proposed in the literature. Since the MCSFilter algorithm features a stochastic component, it was run ten times to solve the NLP problem. The optimization problem was successfully solved in all the runs and the optimal solution demonstrates that in all runs the MCSFilter provides a good quality solution.

Keywords: MCSFilter, α-pinene isomerization model, multistart, derivative-free optimization

1 Introduction

The parameter estimation problem of the α-pinene isomerization model is one of the optimization problems that has been widely used as a benchmark problem to assess the performance of optimization algorithms. This is because of the challenges posed by the multivariable, complex and nonlinear nature of the kinetic model that describes the α-pinene isomerization phenomena. This system is described by the reaction scheme represented in Fig. 1. It comprises five reactions, one of them is reversible, and five chemical species: α-pinene (A), dipentene (B), allo-ocimene (C), α and β-pironene (D), and the dimer (E).
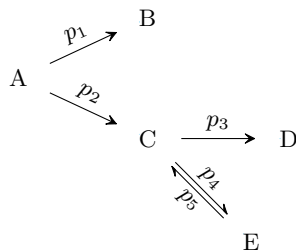
Fig. 1. α-pinene isomerization reaction scheme: A - α-pinene; B - dipentene; C - allo-ocimene; D - α and β-pironene; E - dimer.

One of the earliest kinetic α-pinene isomerization models was proposed by Box and Drapper [1], and it was used to estimate the five kinetic rate constants of this reaction system, using the experimental data obtained by Fuguitt and Hawkins (see [1] and references therein). A later study [2] has demonstrated that there could exist linear dependencies between the experimental data on the concentration of the chemical species involved in the α-pinene isomerization. This was the cause of numerical difficulties to estimate the parameters and the consequent error estimates with respect to the experimental observations. For instance, the concentration of β-pironene was erroneously estimated by 3% of the total conversion of α-pinene [1]. Besides, these linear dependencies might be the root cause of some of the difficulties experienced by optimization methods in getting to the optimal value.

In the work of Ames [3] it is asserted that there are three linear relationships in the experimental chemical species concentration data. The first linear relationship concerns the mass balance to the mixture of the reacting system. The second one is due to the fact that the isomerization is an irreversible phenomenon. Finally, the third linear relationship arises from the fact that the sum of molar fractions of the chemical species in the reacting mixture has to be equal to one. Box et al. [2] solved this parameter estimation problem using a least square criterion. The objective function is defined as the sum of the squared deviation between the chemical species concentrations predicted by the model and the measurements obtained over a given time horizon. Tjoa and Biegler [4] have addressed this estimation problem by solving a constrained nonlinear optimization problem where a quadratic objective function is determined by invoking the numerical integration of the dynamic ODE model of the chemical reaction system. Their results are very similar to those obtained by [2]. In both works, the starting point to solve the optimization problem is very close to the optimal solution, which makes the optimizer convergence easier.

In a followup study based on the earlier works of [5] and [4], Dolan et al. [6] applies a search filter based method that lead to results similar to the ones obtained previously with the derivative based optimization methods. Egea-Larrosa [7,8] has applied the Scatter Search Method (SSm). The SSm is a hybrid populational method that features a stochastic component made of a metaheuristic scatter

search associated with a component that uses the derivative function information. It constitutes a global optimization method with a random local search strategy. A previous formulation of this hybrid method by [9], where it is assumed that the problem is unimodal, was as well as implemented by Egea et al. [7]. The two implementations of the method differ in the local search strategy. The method in [9] uses a local search based on quasi-Newton procedure, whereas in [7] a direct search method is implemented based on a stochastic approach.

A recent work [10], in which the α-pinene system was also used for testing purposes, details the development and implementation of the Firefly Algorithm. This is a stochastic method for global optimization problems that mimics the behavior of fireflies considering that the fireflies are unisex and usually attracted to the brightest light. When applied to the optimization method, the "light intensity" is measured according to the function value: the lower the value, the brightest the light. This means that the points generated by the stochastic part of the method will converge to lower values of the function. For further details on this optimization strategy see [10].

The method used in this work to solve the α-pinene isomerization parameter estimation problem is the Multistart Coordinate Search Filter Method (MCSFilter). It is a derivative-free method based on a multistart strategy coupled with a local coordinate search filter procedure to find the global minimum. Further details on the underlying algorithm are given in Sect. 2 We demonstrate the application of the MCSFilter to solve the α-pinene isomerization parameter estimation problem. Also, a comparison of the performance of the MCSFilter with other optimization strategies is presented regarding the quality of the optimal solution, as well as the influence of two parameters on the performance of the algorithm.

The paper is organized as follows. The derivative-free multistart strategy with coordinate search filter method (MCSFilter) is described in Sect. 2. The kinetic model and the parameter estimation problem are presented in Sect. 3. The numerical results and its discussion are presented in Sect. 4. Finally, some remarks are given in Sect. 5.

2   The Multistart Coordinate Search Filter Method

The MCSFilter algorithm was initially developed by [11] to find multiple solutions of a nonconvex and nonlinear constrained optimization problems of the following type:

$$
\begin{aligned}
\min\ & f(x) \\
\text{subject to}\ & g_j(x) \leq 0, \quad j = 1, ..., m \\
& l_i \leq x_i \leq u_i,\ i = 1, ..., n
\end{aligned}
\tag{1}
$$

where, f is the objective function, $g_j(x)\ j = 1, ..., m$ are the constraint functions and, at least, one of the functions $f, g_j : \mathbb{R}^n \longrightarrow \mathbb{R}$ is nonlinear; l and u are the bounds and $\Omega = \{x \in \mathbb{R}^n : g(x) \leq 0,\ l \leq x \leq u\}$ is the feasible region.

This method does not use any derivative information and incorporate two major different parts: the multistart strategy related with the exploration feature

of the method and a derivative-free local search related with the exploitation of promising regions.

The multistart strategy is a stochastic algorithm that repeatedly applies a local search to sampled points aiming to converge to all the solutions of a multimodal problem. When the direct search is repeatedly applied some minimizers can be found more than once. To avoid a previously computed minimizer, a clustering technique based on computing the regions of attraction of previously identified minimizers is used. In this way, if the sampled point belongs to the region of attraction of an already known minimizer the direct search procedure will not be performed since it would converge to this known minimizer.

Figure 2 illustrates the influence of the regions of attraction. The red/magenta lines between the initial approximation and the minimizer represents a local search that has been performed (red line is used to represent the first local search which converged to each minimum). The white dashed line between the two points represents a local search that was discarded, using the regions of attraction. A set of benchmark problems [11] as well as a small dimensional real problem [12] were used to test the algorithm and the results were very promising.
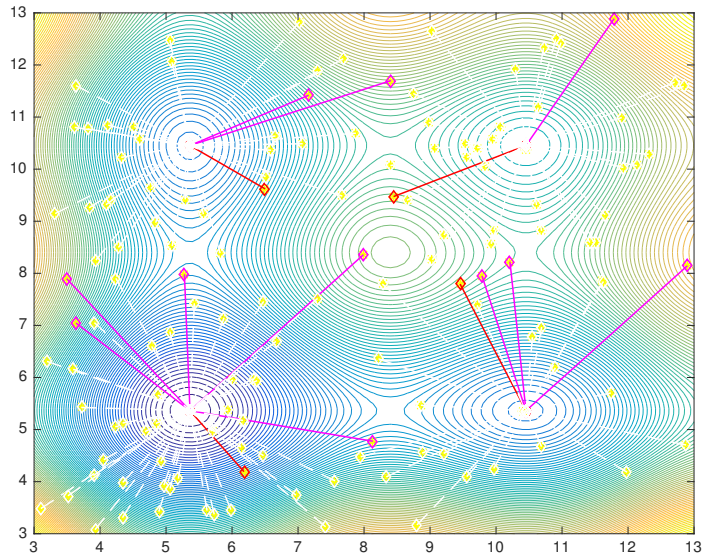
Fig. 2. Illustration of the Multistart strategy with regions of attraction.

The direct search used is a derivative-free local search that consists of a co-ordinate search combined with a filter methodology to generate a sequence of approximate solutions that improve either the constraint violation or the objective function relative to the previous approximation, called Coordinate Search Filter algorithm (CSFilter) [11].

A rough outline of the CSFilter algorithm is as follows and (Algorithm 1 displays the steps of the algorithm). At the beginning, the problem (1) is rewritten as a bi-objective optimization problem aiming to minimize both the objective function $f(x)$ and a nonnegative continuous aggregate constraint violation function $\theta(x)$ defined by

$$\theta(x) = \|g(x)_+\|^2 + \|(l - x)_+\|^2 + \|(x - u)_+\|^2 \tag{2}$$

where $v_+ = \max\{0, v\}$. The filter is initialized to $\mathcal{F} = \{(\theta, f) : \theta \geq \theta_{\max}\}$, where $\theta_{\max} > 0$ is an upper bound on the acceptable constraint violation. The $\mathcal{D}_{\oplus}$ denotes the set of $2n$ coordinate directions, defined as the positive and negative unit coordinate vectors, $\mathcal{D}_{\oplus} = \{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\}$.

Algorithm 1 CSFilter algorithm

Require: x and parameter values, $\alpha_{\min}$; set $\tilde{x} = x$, $x_{\mathcal{F}}^{inf} = x$, $z = \tilde{x}$;
1: Initialize the filter; Set $\alpha = \min\{1, 0.05 \frac{\sum_{i=1}^{n} u_i - l_i}{n}\}$;
2: **repeat**
3:     Compute the trial approximations $z_a^i = \tilde{x} + \alpha e_i$, for all $e_i \in \mathcal{D}_{\oplus}$;
4:     **repeat**
5:       Check acceptability of trial points z_a^i;
6:       **if** there are some z_a^i acceptable by the filter **then**
7:         Update the filter;
8:         Choose z_a^{best}; set $z = \tilde{x}$, $\tilde{x} = z_a^{best}$; update $x_{\mathcal{F}}^{inf}$ if appropriate;
9:       **else**
10:         Compute the trial approximations $z_a^i = x_{\mathcal{F}}^{inf} + \alpha e_i$, for all $e_i \in \mathcal{D}_{\oplus}$;
11:         Check acceptability of trial points z_a^i;
12:         **if** there are some z_a^i acceptable by the filter **then**
13:           Update the filter;
14:           Choose z_a^{best}; Set $z = \tilde{x}$, $\tilde{x} = z_a^{best}$; update $x_{\mathcal{F}}^{inf}$ if appropriate;
15:         **else**
16:           Set $\alpha = \alpha/2$;
17:         **end if**
18:       **end if**
19:     **until** new trial z_a^{best} is acceptable
20: **until** $\alpha < \alpha_{\min}$

The search begins with a central point (the current approximation $\tilde{x}$), and $2n$ trial approximations $z_a^i = \tilde{x} + \alpha e_i$, for $e_i \in \mathcal{D}_{\oplus}$, where $\alpha > 0$ is a step size. The constraint violation value and the objective function value of all $2n$ points are computed. If some trial approximations improve over $\tilde{x}$, reducing θ or f and are acceptable by the filter, then the best of these non-dominated trial approximations, z_a^{best}, is selected, and the filter is updated (adding the corresponding entries to the filter and removing the dominated entries). This best approximation becomes the new central point in the next iteration, $\tilde{x} \leftarrow z_a^{best}$. On the other hand, if all trial approximations z_a^i are dominated by the current filter, then all z_a^i are rejected, and a restoration phase is invoked.

When it is not possible to find a non-dominated best trial approximation (before declaring the iteration unsuccessful) a restoration phase is invoked. In this phase, the most nearly feasible point in the filter, $x_{\mathcal{F}}^{inf}$, is recovered and the search along the $2n$ coordinate directions is carried out from it. If a non-dominated best trial approximation is found, this point becomes the new central point and the iteration is successful. Otherwise, the iteration is unsuccessful, the search returns back to the current $\tilde{x}$, the step size is reduced, $\alpha = \alpha/2$, and new $2n$ trial approximations z_a^i are generated from it. If a best non-dominated trial approximation is still not found, the step size is again reduced since another unsuccessful iteration has occurred. The search stops when α falls below $\alpha_{\min}$, a small positive tolerance. Further details about the multistart strategy and the CSFilter algorithm can be found in [11].

The MCSFilter algorithm was initially coded in MATLAB and, in this work, it was coded in Java language. Algorithm 2 shows the main steps of the MCS-Filter algorithm for finding a global solution to problem (1).

Algorithm 2 MCSFilter algorithm

Require: Parameter values; set $M^* = \emptyset$, $k = 1$, $t = 1$;
1: Randomly generate $x \in [l, u]$; compute $B_{\min} = \min_{i=1,\ldots,n}\{u_i - l_i\}$;
2: Compute $m_1 = \textbf{CSFilter}(x)$, $R_1 = \|x - m_1\|$; set $r_1 = 1$, $M^* = M^* \cup m_1$;
3: **repeat**
4:     Randomly generate $x \in [l, u]$;
5:     Set $o = \arg\min_{j=1,\ldots,k} d_j \equiv \|x - m_j\|$;
6:     **if** $d_o < R_o$ **then**
7:         **if** the direction from x to y_o is ascent **then**
8:             Set $prob = 1$;
9:         **else**
10:            Compute $prob = \varrho\,\phi(\frac{d_o}{R_o}, r_o)$;
11:         **end if**
12:     **else**
13:         Set $prob = 1$;
14:     **end if**
15:     **if** $\zeta^{\ddagger} < prob$ **then**
16:         Compute $m = \textbf{CSFilter}(x)$; set $t = t + 1$;
17:         **if** $\|m - m_j\| > \gamma^* B_{\min}$, for all $j = 1, \ldots, k$ **then**
18:             Set $k = k + 1$, $m_k = m$, $r_k = 1$, $M^* = M^* \cup m_k$; compute $R_k = \|x - m_k\|$;
19:         **else**
20:             Set $R_l = \max\{R_l, \|x - m_l\|\}$; $r_l = r_l + 1$;
21:         **end if**
22:     **else**
23:         Set $R_o = \max\{R_o, \|x - m_o\|\}$; $r_o = r_o + 1$;
24:     **end if**
25: **until** the stopping rule is satisfied

In this algorithm, M^* is the set containing the computed minimizers and ζ is a uniformly distributed number in $(0, 1)$. Moreover, line 17 means that $m \notin M^*$

and in lines 2 and 16, a call is made of the direct search coupled with the filter methodology, the CSFilter algorithm.

The stopping rule that is used in the MCSFilter algorithm is related to the number of initial points used in the multistart strategy. In this way, the algorithm stops when a maximum number of initial points is reached, $k \leq k_{\max}$.

3   Parameter Estimation Problem

As it is illustrated in Fig. 1, the isomerization of the α-pinene is characterized by the formation of dipentene and allo-ocimene, which in turn originates α and β-pironene through an irreversible process reaction, and a dimer through a reversible reaction. Let y_i, $i = 1, \ldots, 5$, denote the molar concentration of each component i, $i = 1, \ldots, 5$, in the mixture, respectively α-pinene, dipentene, allo-ocimene, α and β-pironene, and dimer. The partial mass balance to the mixture leads to the following ODE model:

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} = -(p_1 + p_2)\, y_1, \tag{3a}$$

$$\frac{\mathrm{d}y_2}{\mathrm{d}t} = p_1\, y_1, \tag{3b}$$

$$\frac{\mathrm{d}y_3}{\mathrm{d}t} = p_2\, y_1 - (p_3 + p_4)\, y_3 + p_5\, y_5, \tag{3c}$$

$$\frac{\mathrm{d}y_4}{\mathrm{d}t} = p_3\, y_3, \tag{3d}$$

$$\frac{\mathrm{d}y_5}{\mathrm{d}t} = p_4\, y_3 - p_5\, y_5, \tag{3e}$$

with $t \in [0; 36420]$ and the following initial conditions: $y_1(0) = 100; y_2(0) = 0; y_3(0) = 0; y_4(0) = 0; y_5(0) = 0$. p_j represents the specific reaction rate of reaction j, $j = 1, \ldots, 5$. This is the set of kinetic parameters to be determined in order to fit the model to the experimental data (Table 1).

As mentioned before, because of its complex, nonlinear nature, this model has been used in several works as a benchmark problem to assess the performance of optimization algorithms (e.g., [4,6,8,9,10,13,14,15]).

The parameter estimation problem can be formulated as an optimization problem that applies the least squares method with simple bounds [4]. Since (3) is a dynamic model, the calculation of the value of the objective function requires its numerical integration over the time horizon of interest (see Table 1). The kinetic model parameters, p, are the decision variables of the optimization problem which can be posed as follows:

$$\min_{p} \quad J(p) = \sum_{i=1}^{5} \sum_{k=1}^{9} (y_{\exp i,k} - y_{i,k})^2 \tag{4}$$
$$\text{subject to} \quad p_{\mathrm{L}} \leqslant p \leqslant p_{\mathrm{U}},$$

Table 1. Experimental data (e.g.,[7,10]).

k	t	α-pinene y_1	dipentene y_2	allo-ocimene y_3	pyronene y_4	dimer y_5
1	0	100.0	0.0	0.0	0.0	0.0
2	1230	88.35	7.3	2.3	0.4	1.75
3	3060	76.4	15.6	4.5	0.7	2.8
4	4920	65.1	23.1	5.3	1.1	5.8
5	7800	50.4	32.9	6.0	1.5	9.3
6	10680	37.5	42.7	6.0	1.9	12.0
7	15030	25.9	49.1	5.9	2.2	17.0
8	22620	14.0	57.4	5.1	2.6	21.0
9	36420	4.5	63.1	3.8	2.9	25.7

where $y_{\exp i,k}$ is the experimental concentration value and $y_{i,k}$ the concentration predicted by the model (3) for the chemical specie i at the time instant k. The subscripts U and L on p denote upper and lower limits, respectively.

The best known optimum value of the objective function for this problem is $J(p^*) = 19.872$ achieved at $p_1^* = 5.9256 \times 10^{-5}$, $p_2^* = 2.9632 \times 10^{-5}$, $p_3^* = 2.0450 \times 10^{-5}$, $p_4^* = 2.7473 \times 10^{-4}$, $p_5^* = 4.0073 \times 10^{-5}$ [7].

4   Numerical Results

In this section, the practical performance of the MCSFilter algorithm when solving problem 4 is analyzed. The computational tests were performed on a 2.6 GHz Core i7, with 8 GB of RAM and an operating system MacOs El Capitan laptop. The computational framework to solve the optimization problem (4) was implemented using Java programming language and a Java version of the MCSFilter. During the optimization process, the dynamic model (3) is solved with the explicit Adams-Bashforth integrator for ODEs in Java, from the Apache Commons Math 3.0 Library.

In the simulations done in this study, the absolute and relative tolerances for the integration were set to 1×10^{-8}. The upper and lower bounds on the parameter values were set to:

$$p_{\mathrm{U}} = \begin{bmatrix} 1 \times 10^{-4} & 1 \times 10^{-4} & 1 \times 10^{-4} & 1 \times 10^{-3} & 1 \times 10^{-4} \end{bmatrix}^{\top},$$

$$p_{\mathrm{L}} = \begin{bmatrix} 1 \times 10^{-6} & 1 \times 10^{-6} & 1 \times 10^{-6} & 1 \times 10^{-5} & 1 \times 10^{-6} \end{bmatrix}^{\top}.$$

In order to evaluate the influence of the parameter values ($\alpha_{\min}$ and $k_{\max}$) on the quality of the solution, an experimental study was conducted to tune the stopping criteria parameters of the CSFilter and MCSFilter algorithms, in the context of the α-pinene problem. Nine combinations of $\alpha_{\min}$ and $k_{\max}$, using the values $\alpha_{\min} = \{10^{-5}, 10^{-6}\}$ and $k_{\max} = \{10, 15, 20, 50, 100\}$, were tested.

Table 2 presents the best (J_{best}), average (J_{avg}) and worst (J_{worst}) results produced by MCSFilter algorithm based on 10 executions, as well as the number of function evaluations for the best run, $n_{\text{FE,best}}$.

Table 2. Results obtained by MCSFilter, for different combinations of α_{min} and k_{max}.

α_{min}	k_{max}	J_{best}	J_{avg}	J_{worst}	$n_{\text{FE,best}}$
10^{-5}	10	29.5107	39.8377	75.8033	2933
10^{-5}	15	26.3374	39.6668	83.9623	4601
10^{-5}	20	23.6566	33.6128	52.1003	2729
10^{-5}	50	22.9355	30.6804	44.8241	14026
10^{-6}	10	20.0165	20.3306	20.9549	5304
10^{-6}	15	19.9517	20.1647	20.6374	5249
10^{-6}	20	19.9122	20.1013	20.3227	8486
10^{-6}	50	19.9157	20.0173	20.1143	20737
10^{-6}	100	19.9074	19.9798	20.1241	34241

Table 2 shows that MCSFilter algorithm converges to the known global minimum. The objective of this experimental study was to evaluate the performance of the MCSFilter algorithm for different values of α_{min} in order to obtain accurate solutions, but with a low number of function evaluations. However, despite the reduced number of evaluations of the function for $\alpha_{\text{min}} = 10^{-5}$ the obtained solutions are not good with a relative error of 16 % or more of the global optimum known solution. It follows that an increase in the value of the parameter α_{min} leads to worst solutions since the global solution was never reached. In addition, we can also conclude that as k_{max} increases the number of evaluations of the function also increases. This is obvious, because if the number of points generated is greater, the number of evaluations of the function is also greater.

The best value of $J(p) = 19.9074$ is obtained for $\alpha_{\text{min}} = 10^{-6}$ and $k_{\text{max}} = 100$ with a relative error of $+0.17\%$ when compared with the best known solution in [7]. From Table 2 one may conclude that for smaller values of α_{min} the solution is better. Thus, increasing the value of α_{min} leads to a worse solution than reducing the number of initial points.

Finally, a last experiment with one execution of the MCSFilter algorithm was performed using $\alpha_{\text{min}} = 10^{-6}$, $k_{\text{max}} = 100$ and the following first initial point (in the multistart part):

$$p = \begin{bmatrix} 6.05 \times 10^{-5} & 3.60 \times 10^{-5} & 4.18 \times 10^{-5} & 4.79 \times 10^{-4} & 9.86 \times 10^{-5} \end{bmatrix}^{\top} .$$

The optimal objective function value found was $J(p) = 19.8828$. Since this result is very close to the best known optimum by Egea et al. [7], one can assert that an execution with a local search starting close to the optimum allows to obtain a very good solution, in the fastest way as stated by [5]. This means that, in all the experiments, the points sampled in the multistart stage are farther away

from the known minimizer than the above initial point; in spite of that, and for $\alpha_{\min} = 10^{-6}$, the MCSFilter algorithm always obtained high quality solutions.

The results obtained with the MCSFilter were compared with those obtained with other methods [7,8,10]. Table 3 aggregates the published best found solution, J_{best}, the average solution (when existent), J_{avg}, and the average number of function evaluations, $n_{\text{FE,avg}}$, obtained for some methods in the literature. This table also presents the values for MCSFilter algorithm related with the 9 executions starting from a random initial point plus the execution with the first initial point closed to the optimum, as described above. The average values presented for DE, SRES, SSm, FA, and MCSFilter are over 10 executions.

Table 3. α-pinene isomerization parameter estimation problem results reported in the literature.

Solver	J_{best}	J_{avg}	$n_{\text{FE,avg}}$
MCSFilter	19.8828	19.9703	40494
Global[7]	31638	35225	1277
DE [8]	34.856	22515	10000
SRES [8]	31251	32651	10000
DIRECT[8]	36218	-	9996
OQNLP [8]	31252	-	10000
SSm [8]	19.872	19.872	9518
SSm [7]	19.872	24.747	1163
FA [10]	19.8772	25.6777	5860[a]
fmincon[10]	19.929	-	217

[a] Number of function evaluations of the best solution.

The methods Global, DE, SRES, DIRECT and OQNLP fail to converge to the optimum value because of the limit (10000) on the maximum number of function evaluations [8] and, therefore, cannot be compared to the value obtained by the other methods. Besides that it is remarked that with $\alpha_{\min} = 10^{-6}$ and a smaller number of function evaluations (less than 10000 — see Table 2) the MCSFilter algorithm achieved a very good results.

Regarding the number of function evaluations, it can be observed that, in spite of the good quality of the solution and the MCSFilter consistency, the algorithm requires more function evaluations than the other methods listed in Table 3. Nevertheless, the MCSFilter presents better average values of the objective function than SSm [7] and FA [10], and with a relative error of +0.54%. This demonstrates that the MCSFilter has converged more often to the best solution than some other methods.

We remark that the best MCSFilter solution is close to its average solution (see Table 2), which are close to the known best minimum. These results assert the good quality of the solution presented by the MCSFilter and its consistency.

Table 4 shows the minimizer of the best solution obtained by MCSFilter for $\alpha_{\min} = 10^{-6}$ and $k_{\max} = 100$ that corresponds to the best and worst solutions presented in Table 2.

Table 4. Parameter values obtained by MCSFilter for $\alpha_{\min} = 10^{-6}$ and $k_{\max} = 100$.

parameter	best	worst
p_1^*	5.936×10^{-5}	5.904×10^{-5}
p_2^*	2.965×10^{-5}	2.969×10^{-5}
p_3^*	2.067×10^{-5}	1.978×10^{-5}
p_4^*	2.776×10^{-4}	2.835×10^{-4}
p_5^*	4.090×10^{-5}	4.448×10^{-5}

Figure 3 illustrates the profiles obtained by solving the ODE model (3) using the parameter set, p^*, found by the MCSFilter algorithm, for the best solution in Table 4.
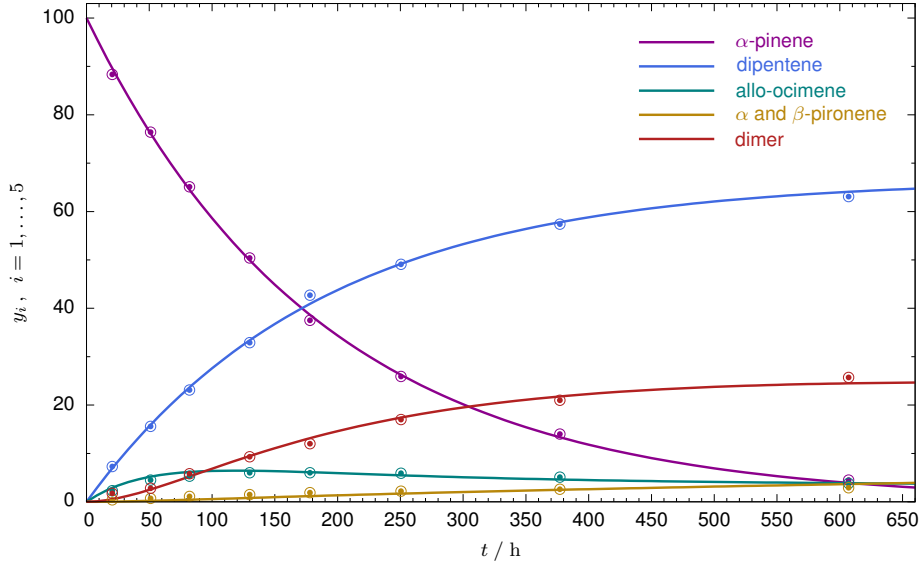
Fig. 3. Experimental data points and the profiles obtained from (3) with the kinetic model parameters determined by the MCSFilter.

As it can be observed, the obtained kinetic parameters provide a good fitting of the model to the experimental data.

5   Conclusions

The MSCFilter algorithm — a multistart strategy coupled with a coordinate search filter methodology — was implemented in the Java programming language and has successfully estimated the α-pinene isomerization optimal kinetic parameters.

Some preliminary experiments were carried out in order to tune the parameters of the stopping criteria of the CSFilter and MCSFilter algorithms. It is shown that this issue is crucial to the effective convergence of the algorithm when solving the α-pinene problem.

In comparison with the results reported in the literature, it is noticeable that the MCSFilter presents a much higher effort when 100 initial points are used. This is due to the higher number of function evaluations required by the MCSFilter method, namely by the CSFilter algorithm based on the set $\mathcal{D}_{\oplus}$ with $2n = 10$ coordinate directions when it is using 100 initial points. One may conclude that increasing $\alpha_{\min}$ has a big impact: CSFilter stops before reaching the best known solution. Despite this, the MCSFilter algorithm shows good performance by converging to the known solution of the problem — the experiments that were carried out show that (when $\alpha_{\min} = 10^{-6}$ is used) the best, worst and the average values of the objective function obtained by MCSFilter algorithm are closer to the best known in the literature.

The fitting of the model profiles to the experimental data demonstrate that the obtained solution is of very good quality.

As future work, a strategy of parallelization inside the MCSFilter algorithm, namely in the multistart strategy stage, should be implemented. The MCSFilter algorithm will also be applied in the context of dynamic systems where its ability to handle inequality constraints is of importance.

References

1. Box, G. E. P., Draper, N. R.: The bayesian estimation of common parameters from several responses. Biometrika, 52(3-4), 355-365 (1965).
2. Box, G. E. P., Hunter, W. G., MacGregor, J. F., Erjavec, J.: Some problems associated with the analysis of multiresponse data. Technometrics, 15(1), 33-51 (1973).
3. Ames, W. F.: Canonical forms for non-linear kinetic differential equations. Industrial & Engineering Chemistry Fundamentals, 1(3), 214-218 (1962).
4. Tjoa, I.-B., Biegler, L. T.: Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. Industrial & Engineering Chemistry, 30, 376-385 (1991).
5. Averick, B. M., Carter, R. G., Mor, J. J., Xue, G.: The minpack-2 test problem collection. Technical Report, Mathematics and Computer Science Division, Argonne National Laboratory (1992).
6. Dolan, E. D., Mor, J. J., Munson, T. S.: Benchmarking optimization software with cops 3.0. Technical Report, Argonne National Laboratory (2004).
7. Egea, J. A., Rodriguez-Fernandez, M., Banga, J. R., Mart, R.: Scatter search for chemical and bio-process optimization. Journal of Global Optimization, 37(3), pp. 481-503 (2007).

8. Larrosa, J. A. E.: New Heuristics for Global Optimization of Complex Bioprocesses. PhD Thesis, University of Vigo (2008).

9. Csendes, T.: Non-linear parameter estimation by global optimization - efficiency and reliability. Acta Cybernetica, 8(4), 361-370 (1988).

10. Rocha, A. M. A. C., Martins, M. C., Costa, M. F. P., Fernandes, E. M. G. P.: Direct sequential based firefly algorithm for the α-pinene isomerization problem. In Gervasi, O., Murgante, B. et al. (eds) Computational Science and Its Applications — ICCSA 2016. LNCS, 9786, 386-401. Springer Berlin Heidelberg (2016).

11. Fernandes F. P., Costa M. F. P., Fernandes, E. M. G. P.: Multilocal Programming: A Derivative-Free Filter Multistart Algorithm. In: Murgante B. et al. (eds) Computational Science and Its Applications — ICCSA 2013. LNCS, 7971, 103-118. Springer, Berlin, Heidelberg (2013).

12. Amador, A., Fernandes, F. P., Santos, L. O., Romanenko, A.: Application of MCS-Filter to estimate stiction control valve parameters. International Conference of Numerical Analysis and Applied Mathematics, AIP Conference Proceedings, 1863, pp. 270005 (2017).

13. Storn, R., Price, K.: Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11, 341-359 (1997).

14. Runarsson, T. P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. Institute of Electrical and Electronics Engineers Transactions on Evolutionary Computation, 4(3), 284-294 (2000).

15. Jones, D. R.: Direct global optimization algorithm. In Encyclopedia of Optimization, Floudas, C. A. and Pardalos, P. M., editors, 431-440. Springer US, Boston, MA (2001).

16. Laguna, M. and Martí, R.: Scatter Search: Methodology and Implementations in C. Kluwer Academic Publishers, Boston (2003).